



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

bacon david compiler optimization

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#)

bacon david compiler optimization

Terms used: [bacon david compiler optimization](#)

Found 216 of 238,273

Sort results
by

relevance

[Save results to a Binder](#)

Refine these results with [Advanced Search](#)

Display
results

expanded form

☐ Open results in a new
window

Try this search in [The ACM Guide](#)

Results 1 - 20 of 216

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#) [>>](#)

- 1 [On increasing architecture awareness in program optimizations to bridge the gap between peak and sustained processor performance: matrix-multiply revisited.](#)

David Parelo, Olivier Temam, Jean-Marie Verdun

November 2002 **Supercomputing '02**: Proceedings of the 2002 ACM/IEEE conference on Supercomputing

Publisher: IEEE Computer Society Press

Additional Information: [full citation](#), [abstract](#),

Full text available: pdf(263.32 KB)

[references](#), [cited by](#), [index terms](#)

Document
Scanning Service
Free Online Quote.
Scan to PDF/TIF
Serving the DC
Metropolitan Area

As the complexity of processor architectures increases, there is a widening gap between peak processor performance and sustained processor performance so that programs now tend to exploit only a fraction of available performance. While there is a tremendous ...

- 2 [Proceedings of the 2006 ACM SIGPLAN/SIGBED conference on Language, compilers, and tool support for embedded systems](#)



Mary Jane Irwin, Koen De Bosschere

June 2006 proceeding

Publisher: ACM

Additional Information: [full citation](#), [abstract](#)

It is our great pleasure to welcome you to the *ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems -- LCTES 2006*. This year's conference continues its tradition of being the premier forum for presentation of research ...


- 3 [Compiler-directed page coloring for multiprocessors](#)



Edouard Bugnion, Jennifer M. Anderson, Todd C. Mowry, Mendel Rosenblum, Monica S. Lam

October 1996 **ASPLOS-VII**: Proceedings of the seventh international conference on Architectural support for programming languages and operating systems

Publisher: ACM

Full text available:  [pdf\(1.37 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)


This paper presents a new technique, *compiler-directed page coloring*, that eliminates conflict misses in multiprocessor applications. It enables applications to make better use of the increased aggregate cache size available in a multiprocessor. ...

4 A comparison of empirical and model-driven optimization



Kamen Yotov, Xiaoming Li, Gang Ren, Michael Cibulskis, Gerald DeJong, Maria Garzaran, David Padua, Keshav Pingali, Paul Stodghill, Peng Wu
June 2003 **PLDI '03**: Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation

Publisher: ACM

Full text available:  [pdf\(448.74 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Empirical program optimizers estimate the values of key optimization parameters by generating different program versions and running them on the actual hardware to determine which values give the best performance. In contrast, conventional compilers ...


Keywords: BLAS, blocking, code generation, compilers, empirical optimization, memory hierarchy, model-driven optimization, program transformation, tiling, unrolling

5 Compiler-directed page coloring for multiprocessors



Edouard Bugnion, Jennifer M. Anderson, Todd C. Mowry, Mendel Rosenblum, Monica S. Lam
December 1996 **ASPLOS-VII: ACM SIGOPS Operating Systems Review**, Volume 30 Issue 5

Publisher: ACM


Full text available:  [pdf\(1.37 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

This paper presents a new technique, *compiler-directed page coloring*, that eliminates conflict misses in multiprocessor applications. It enables applications to make better use of the increased aggregate cache size available in a multiprocessor. ...

6 Value reuse optimization: reuse of evaluated math library function calls through compiler generated cache



K. V. Seshu Kumar
August 2003 **ACM SIGPLAN Notices**, Volume 38 Issue 8
Publisher: ACM

Full text available:  [pdf\(880.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#)

Value reuse technique eliminates the redundant evaluation of expressions, using the support of hardware at runtime to eliminate them. The potential performance of a value reuse mechanism not only

depends on the number of instances it has eliminated, ...

Keywords: Compilers Optimization, Function Cache, Function reuse, Instruction reuse

7 Compiler optimizations for nondeferred reference: counting garbage collection



Pramod G. Joisha

June 2006 **ISMM '06**: Proceedings of the 5th international symposium on Memory management

Publisher: ACM

Full text available: [pdf\(220.00 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Reference counting is a well-known technique for automatic memory management, offering unique advantages over other forms of garbage collection. However, on account of the high costs associated with the maintenance of up-to-date tallies of references ...

Keywords: reference counting, static analyses

8 A study of devirtualization techniques for a Java Just-In-Time



compiler

Kazuaki Ishizaki, Motohiro Kawahito, Toshiaki Yasue, Hideaki Komatsu, Toshio Nakatani

October 2000 **OOPSLA '00**: Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications

Publisher: ACM

Full text available: [pdf\(225.89 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

Many devirtualization techniques have been proposed to reduce the runtime overhead of dynamic method calls for various object-oriented languages, however, most of them are less effective or cannot be applied for Java in a straightforward manner. This ...

9 Proceedings of the 2007 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools



Santosh Pande, Zhiyuan Li

June 2007 proceeding

Publisher: ACM

Additional Information: [full citation](#), [abstract](#)

It is with great pleasure that we welcome you to the *ACM 2007 Conference on Languages Compilers and Tools for Embedded Systems (LCTES'07)* on behalf of its organizational committees. The aim of LCTES is to provide a premier forum for sharing the ...

10 Proceedings of the 2005 ACM SIGPLAN/SIGBED conference onLanguages, compilers, and tools for embedded systems

Yunheung Paek, Rajiv Gupta

June 2005 proceeding

Publisher: ACMAdditional Information: [full citation](#), [abstract](#)

It is our great pleasure to welcome you to the *ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems* -- *LCTES'05*. This year's conference continues its tradition of being the premier forum for presentation of research ...

11 Design, implementation, and evaluation of optimizations in a just-in-time compiler

Kazuaki Ishizaki, Motohiro Kawahito, Toshiaki Yasue, Mikio Takeuchi, Takeshi Ogasawara, Toshio Suganuma, Tamiya Onodera, Hideaki Komatsu, Toshio Nakatani

June 1999 **JAVA '99**: Proceedings of the ACM 1999 conference on Java Grande**Publisher:** ACMFull text available: [pdf\(1.09 MB\)](#) Additional Information: [full citation](#), [references](#), [cited by](#), [index terms](#)**12** A compiler framework for restructuring data declarations to enhance cache and TLB effectiveness

David F. Bacon, Jyh-Herng Chow, Dz-ching R. Ju, Kalyan Muthukumar, Vivek Sarkar

October 1994 **CASCON '94**: Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research**Publisher:** IBM PressFull text available: [pdf\(298.15 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

It has been observed that memory access performance can be improved by restructuring data declarations, using simple transformations such as array dimension padding and inter-array padding (array alignment) to reduce the number of misses in the cache ...

13 Loop fusion for memory space optimization

Antoine Fraboulet, Karen Kodary, Anne Mignotte

September 2001 **ISSS '01**: Proceedings of the 14th international symposium on Systems synthesis**Publisher:** ACMFull text available: [pdf\(152.91 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)


Portable or embedded systems as well as submicronic technologies have made the power consumption criterium crucial. Memory is known to be

extremely power consuming. Moreover multimedia applications are memory intensive applications. Therefore, we propose ...

14 A study of devirtualization techniques for a Java Just-In-Time

compiler

Kazuaki Ishizaki, Motohiro Kawahito, Toshiaki Yasue, Hideaki Komatsu, Toshio Nakatani
October 2000 **OOPSLA '00: ACM SIGPLAN Notices**, Volume 35 Issue 10
Publisher: ACM


Full text available:  [pdf\(225.89 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

Many devirtualization techniques have been proposed to reduce the runtime overhead of dynamic method calls for various object-oriented languages, however, most of them are less effective or cannot be applied for Java in a straightforward manner. This ...

15 Stack allocation and synchronization optimizations for Java using

escape analysis

Jong-Deok Choi, Manish Gupta, Mauricio J. Serrano, Vugranam C. Sreedhar, Samuel P. Midkiff
November 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 25 Issue 6
Publisher: ACM

Full text available:  [pdf\(632.85 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#), [review](#)


This article presents an *escape analysis* framework for Java to determine (1) if an object is not reachable after its method of creation returns, allowing the object to be allocated on the stack, and (2) if an object is reachable only from a single ...

Keywords: Connection graphs, escape analysis, points-to graph

16 Data size optimizations for java programs



C. Scott Ananian, Martin Rinard
July 2003 **LCTES '03: ACM SIGPLAN Notices**, Volume 38 Issue 7
Publisher: ACM

Full text available:  [pdf\(349.36 KB\)](#) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

We present a set of techniques for reducing the memory consumption of object-oriented programs. These techniques include analysis algorithms and optimizations that use the results of these analyses to eliminate fields with constant values, reduce the ...

Keywords: bitwidth analysis, embedded systems, field externalization, field packing, size optimizations, static specialization

17 A comparison of empirical and model-driven optimization

Kamen Yotov, Xiaoming Li, Gang Ren, Michael Cibulskis, Gerald DeJong, Maria Garzaran, David Padua, Keshav Pingali, Paul Stodghill, Peng Wu
May 2003 **PLDI '03: ACM SIGPLAN Notices**, Volume 38 Issue 5

Publisher: ACM

Full text available: pdf(448.74 KB) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

Empirical program optimizers estimate the values of key optimization parameters by generating different program versions and running them on the actual hardware to determine which values give the best performance. In contrast, conventional compilers ...

Keywords: BLAS, blocking, code generation, compilers, empirical optimization, memory hierarchy, model-driven optimization, program transformation, tiling, unrolling

18 Data size optimizations for java programs

C. Scott Ananian, Martin Rinard
June 2003 **LCTES '03: Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems**

Publisher: ACM

Full text available: pdf(349.36 KB) Additional Information: [full citation](#), [abstract](#),
[references](#), [cited by](#), [index terms](#)

We present a set of techniques for reducing the memory consumption of object-oriented programs. These techniques include analysis algorithms and optimizations that use the results of these analyses to eliminate fields with constant values, reduce the ...

Keywords: bitwidth analysis, embedded systems, field externalization, field packing, size optimizations, static specialization

19 Compiler optimization-space exploration

Spyridon Triantafyllis, Manish Vachharajani, Neil Vachharajani, David I. August

March 2003 **CGO '03: Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Publisher: IEEE Computer Society

Full text available: pdf(1.19 MB) Additional Information: [full citation](#), [abstract](#), [references](#),
[cited by](#), [index terms](#)

To meet the demands of modern architectures, optimizing compilers must incorporate an ever larger number of increasingly complex transformation algorithms. Since code transformations may often degrade performance or interfere with subsequent transformations, ...

20 Compiler transformations for high-performance computing

David F. Bacon, Susan L. Graham, Oliver J. Sharp



December 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 4

Publisher: ACM

Full text available: [pdf\(6.32 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#),
[cited by](#), [index terms](#), [review](#)

In the last three decades a large number of compiler transformations for optimizing programs have been implemented. Most optimizations for uniprocessors reduce the number of instructions executed by the program using transformations based on the analysis ...

Keywords: compilation, dependence analysis, locality, multiprocessors, optimization, parallelism, superscalar processors, vectorization

Results 1 - 20 of 216

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#) [>>](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2008 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)